

グリッドにおける データベースアクセス & 統合 (WS-DAI & OGSA-DAI)

産総研グリッド研究センター

データグリッドチーム

小島 功

kojima@ni.aist.go.jp

<http://www.dbgrid.org>



概要

- **グリッドにおけるデータベース処理とは？（基礎的内容）**
 - データグリッドにおけるデータベースアクセス
 - 応用の紹介・機能上の特徴など
- **OGSA-DAIとは**
 - 英国のグループを中心としたミドルウェア実装
 - OGF(Open Grid Forum)における標準活動とリンクした活動
- **OGSA-DAIの詳細**
 - 機能の説明
- **OGF WS-DAIの概要**
 - OGF (Global Grid Forum)における標準化活動
- **OGSA-DQPの概要**
- **OGSA-DAI 3.0 の概要**

背景

- **情報の爆発的増加**
 - － 応用での情報量が増えている
 - 秒あたりGBの情報量を出すセンサー、
 - 100TBのCG映画(Pixar)
 - 膨大なネット上の情報など
 - － ディスクが安い
- **格納形式はさまざま**
 - － DB,ファイルなど
 - － XML,HTML,エクセルにいたるまで
- **分散環境でよい統合・連携の枠組みが必要となる**
 - － 分散環境 ⇒ **グリッド**
 - ★ スケーラビリティ(パフォーマンス&分散)
 - ★ 異種のプロダクト・製品の組み合わせ
 - － データのよい統合・連携の枠組み？

データ・グリッド

- グリッド応用における、データ指向計算とは
 - 大量のデータを扱う分散計算
 - CERN のセンサデータの分散配布・共有・レプリカ管理
 - 衛星データの配信など
 - 地理的・組織的に分散した情報の統合計算
 - IVO(仮想天文台)に基づくデータベース天文学
 - 統合遺伝子データベース上のデータ解析・マイニングなど

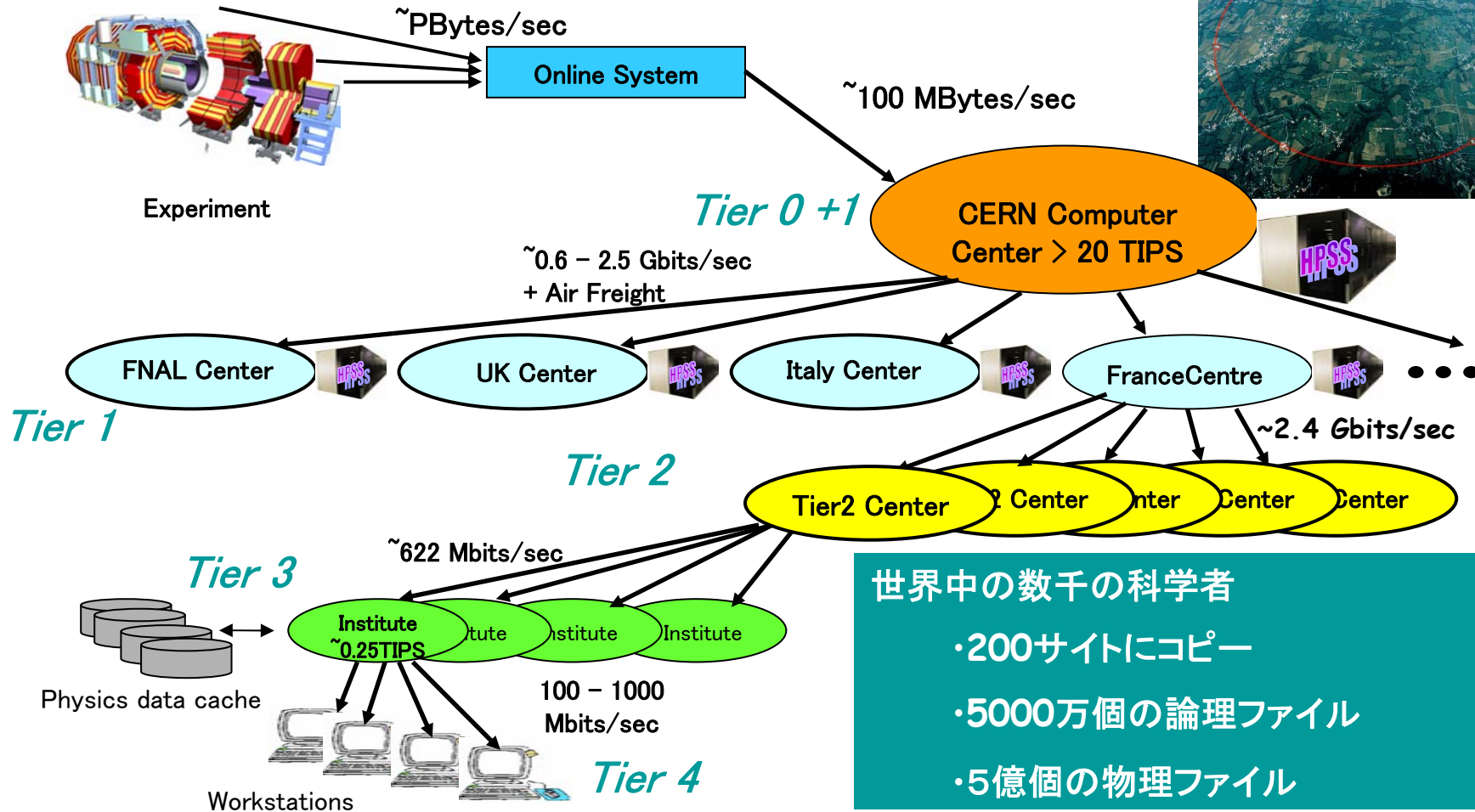
これらの応用やインフラを広くデータ・グリッドと呼ぶ。

データグリッド応用の種類 分散・大量データ処理(トップダウン的)

- CERN-LHC, 衛星データ, センサーネットワークなど
- **ひとつの情報リソースからのデータが膨大**
 - 単一のストレージに入らない
 - 同一タイプのデータソースが膨大にある
 - 分散した解析など、分散環境でデータを共有したい。
- **単一の管理ポリシーのデータをトップダウン的に分散させる技術。**

動機的な例 CERN LHC

100MB/s -> 360GB/H -> 8.64TB/Day -> 3PB/Year



(注: 2001 段階)

データグリッド応用の種類 情報統合(ボトムアップ的)

- 遺伝子データベースの統合と解析
- 仮想天文台など
- 分散した地域、組織などから、
データがボトムアップ的に発生する。
 - 意味が似ているのに表現が違う。
 - 似たようなデータを持っている。
 - 組織独自のデータ管理ポリシーを維持したい
- 異なる組織などで別々に作られたデータを、
仮想的にひとつのデータに統合する(情報統合)

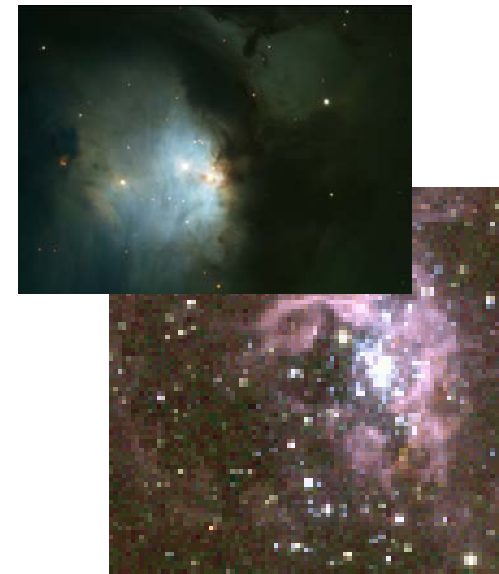
動機的な例2: 仮想天文台 & データベース天文学

望遠鏡データ: テラバイト〜ペタバイト

- **2MASS Project (2Micron All Sky Survey)**
 - 赤外線光での全天のサーベイ
 - 500万イメージ・10テラバイト
- **SDSS (Sloan Digital Sky Survey)**
 - 全天の4分の1 紫外〜近赤外
 - 40テラバイトの生データ
 - 600GBのカタログ
- 他、LIGO (重力波観測レーザー干渉計) など

Virtual Observatory

- 分散した複数のサーベイデータを統合して扱いたい。
- 多様な測定データを組み合わせる解析したい。
- 分散データ・マイニング
- 空間データ検索
(近傍のオブジェクトとか、移動オブジェクトの検索とか)



どう分散を吸収し、仮想化するか？

分散の吸収: 仮想化のレベルや応用が違う。(組み合わせることも可能)

- 分散ファイルシステム
 - Gfarmなど、単一の管理ドメインを広げる応用などに向く。
- 分散データベース
 - OGSA-DAIなど 特に、情報統合的な側面を持つ応用に有益

なぜか？

データの抽象表現 (テーブル) => 異なる情報や応用をつなぐための共通表現
物理表現/構造に独立 => 環境の違いを吸収できる。

抽象的なデータ表現 (ex. テーブル) によるデータ管理は、応用依存性が少なく有益。

プログラム言語に依存しない

SQLなど宣言的な検索言語は大量データ処理に向く
トランザクション処理、一貫性、権限などDBMS機能

- その他
 - Gridftp、ftpなどデータ転送ツール
 - レプリカ管理など、分散に伴うデータの一貫性保持ツール

これらの機能群、ツール群を、よく連携させること。

グリッドにおけるデータベース処理の特徴(1)

- サービスベースの分散データベースアクセス
 - Webサービスの規格を通して遠隔のデータベースをアクセスする。
RDB,XMLDB、OODBなど。

Webサービス経由のJDBCとかでいいのでは？

NO !

- 検索などデータの量が大量であること。
 - 結果をファイル転送(FTP)などで戻したい。
- 複数のサイトで分散処理
 - 分散問い合わせ。
 - 第3者転送。(検索結果を別のサイトに送るなど)

グリッドにおけるデータベース処理の特徴(2)

- 単なる結果データの転送だけでない処理
 - 転送のついでにフォーマット変換とかしたいが、いちいちファイルに書くのは嫌だ。
 - 大量なので圧縮して戻したい。
- グリッドの他のツールとの親和性
 - GSI(グリッドにおける標準の認証)の支援: 計算処理との連携・融合
 - 管理ツールやインターフェイスなどのサービスレベルの互換性
- 異なるデータリソースの扱いを統合・連携したい
 - ファイル処理
 - WebDBなど

これらを満足するDB規格とミドルウェアがほしい

グリッド(OGSA)における規格とミドルウェア

- **OGSA(Open Grid Service Architecture)**
 - サービスベースのグリッドのアーキテクチャ
 - OGF(Open Grid Forum)で策定
- **OGSAに基づくデータベースアクセス規格**
 - **WS-DAI**
 - Web Service Database Access and Integration
 - OGF DAIS Working Groupで策定
 - OASISやW3Cにおける、WS-XXに対応
- **WSRF(GT4)に基づく、データベースミドルウェア**
 - **OGSA-DAI**
 - GT4(など)上でデータベースをアクセスするためのミドルウェア
- **特徴:この2つが密に連携している。**
 - WGのメンバーが開発に関与。
 - 開発のフィードバックを仕様に反映、
 - OGSA-DAIがDAIS仕様の参照実装になる、、などなど。
- **標準化活動との密接な関連 = 使える標準へ**

OGSA-DAI とは

OGSA-DAIとは

- Webサービスを基礎とした、データベースアクセスのミドルウェア
 - OGSA(Open Grid Service Architecture)
 - DAI(Database Access and Integration)

単純には、リモートのDBをWebサービス経由でアクセスするミドルウェア(もちろんそれだけではないが)

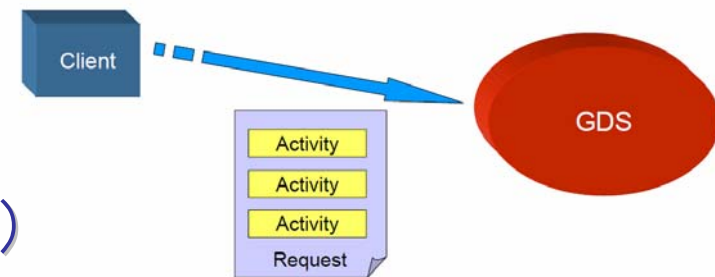
英国 Uk-e-Scienceプログラムの1プロジェクトとしてスタート
現在は、OMII (Open Middleware Infrastructure Institute)の
1プロジェクト <http://www.omii.ac.uk/>

- OMII グリッドのミドルウェア
- OGSA-DAI データベース
- myGrid/Taverna セマンティックWeb・ワークフロー

OGSA-DAIでできることは?

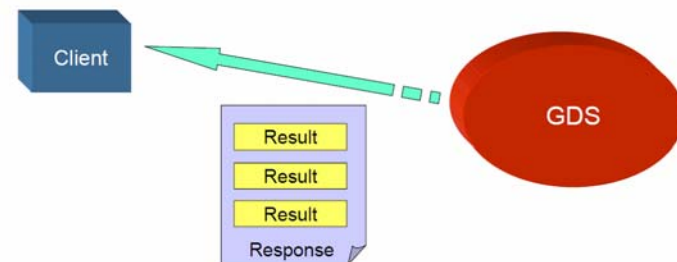
●リモート・遠隔地のデータベースが、Webサービス(データサービス、GDS)として提供され、グリッドのミドルウェア(Globus Toolkit)上のプログラム(client)から使える。

➤一般的なデータベースプログラミング
(プログラム内からSQLなどを投げる)



●アクティビティという処理モデル(後述)

➤一連の処理をまとめて投げ、
➤処理結果をまとめてもらう。



★ OGSA-DAIの考え方: 分散処理へのアプローチ

● やりたいこと

- 分散データベース処理を含む分散ワークフロー
 - サービス連携
 - SOAベースの応用開発

ナイーブに応用を作ると性能が悪くなりそうだ。

★ OGSA-DAIにおける考え方。

1. 一つのデータサービスでの処理を高度化する。

- Activityの連携でワークフローを記述、処理
 - ひとつのサービス内でできることを高度化

2. 複数のサービスにおける連携(後述)

- データベース処理を含む汎用ワークフロー
 - BPELやTavernaなど、汎用ワークフローエンジンと組み合わせよ。
 - 該当モジュールの提供
 - WEEPなどDAI向きBPELエンジンの提供
- 分散データベース処理に基づくワークフロー
 - 分散問い合わせ処理のミドルウェア+ α の提供
 - OGSA-DQP
 - 問い合わせ処理を最適化する。

いいところ

単なるSQLのリモートアクセスとどう違うか？

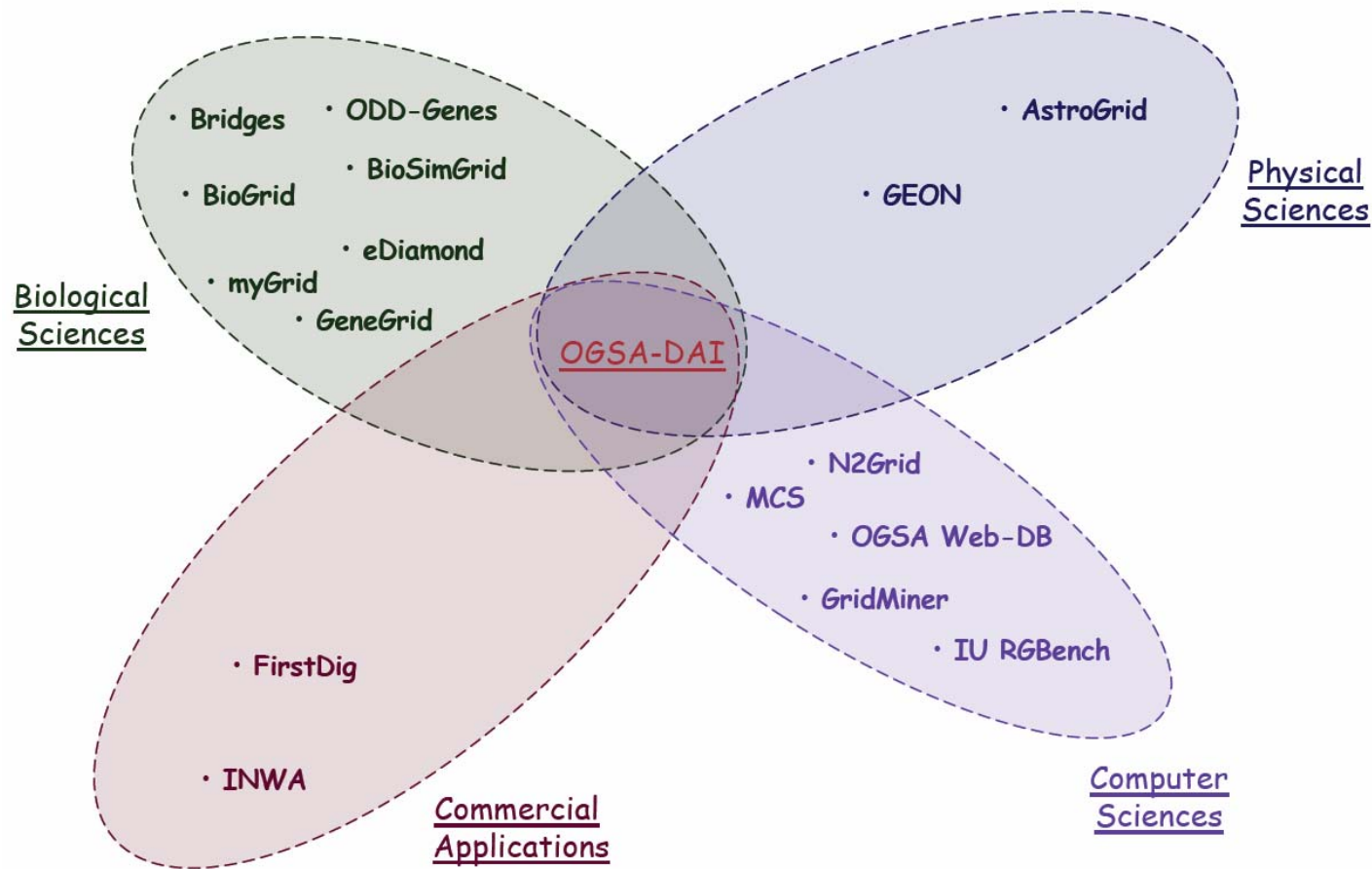
- サービスベース
 - HTTPポートの開いているところでは、どこでもOK
- GSIをサポート
 - GSIとDB(例えばOracle)のアカウントのマッピング
- 結果のデータ変換をサポート
 - XSLTを支援、検索結果の加工など。
- 第3者転送・大量データ処理をサポート
 - FTP・GFTPなどにより、結果を第3者サイトに転送できる

実装上の制限

- **分散問い合わせやサービス連携:別のミドルの分担**
 - 関係DBについての分散問い合わせができる。
 - OGSA-DQP(最適化も含め実行計画を立ててくれる)
 - XMLの分散問い合わせはない。
 - それ以上の処理はワークフローエンジンと組み合わせる
- **ミドルウェア→プロダクト依存性がある**
 - 問い合わせを投げ、結果をもらうためのミドルウェア
 - XMLDBとRDBを問い合わせレベルでは統合できない。
 - RDBのプロダクト固有の機能については、連携・統合できない。

どのような利用があるか？

たくさん利用・応用・開発プロジェクト

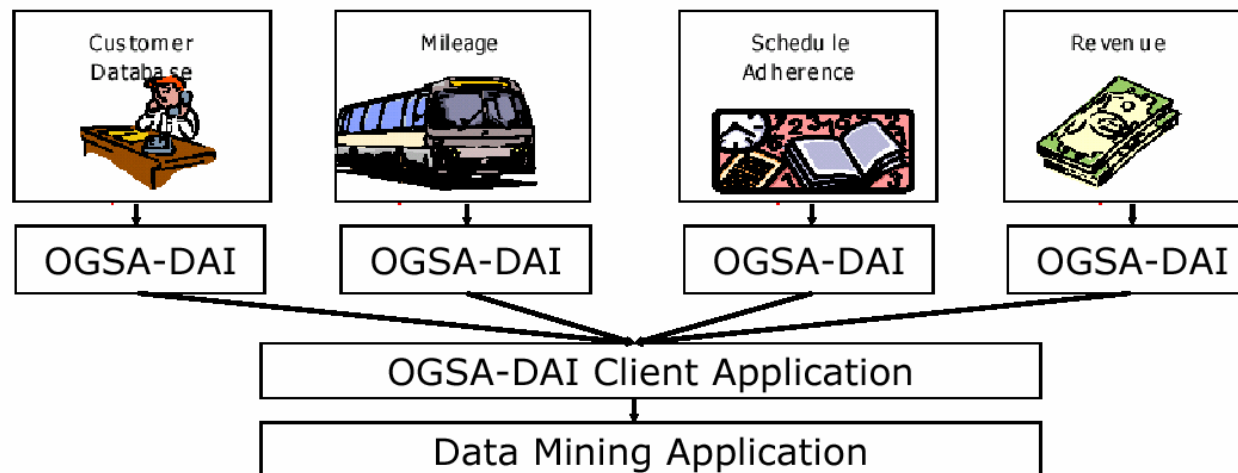


ユーザグループ: ogsadai.org.uk

どんな風に使われているか。

● 例: FirstDIG

- 商用(交通サービス、FirstTransportGroup)利用でのOGSA-DAI
- 複数のデータベースにまたがったデータマイニング
- バスが10分以上遅れると82%の確率で収入が10%下がる、とか



どんな風に使われているか？

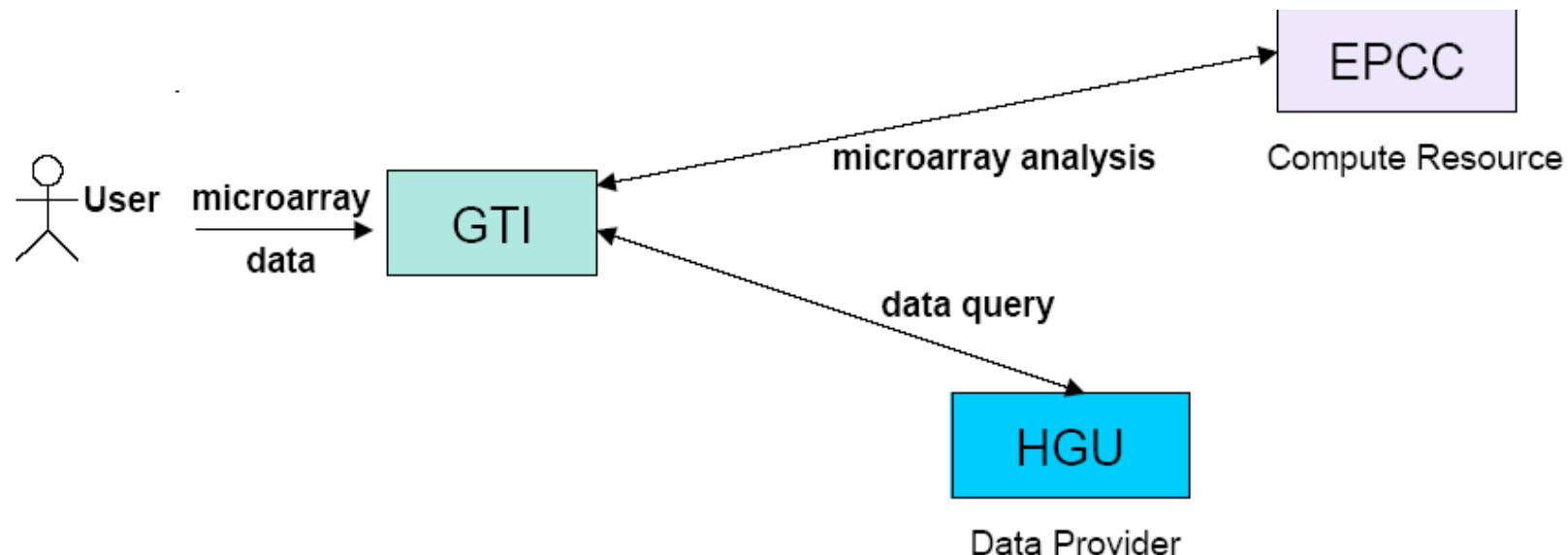
ODD-Genes

OgsaDai Demo for Genes

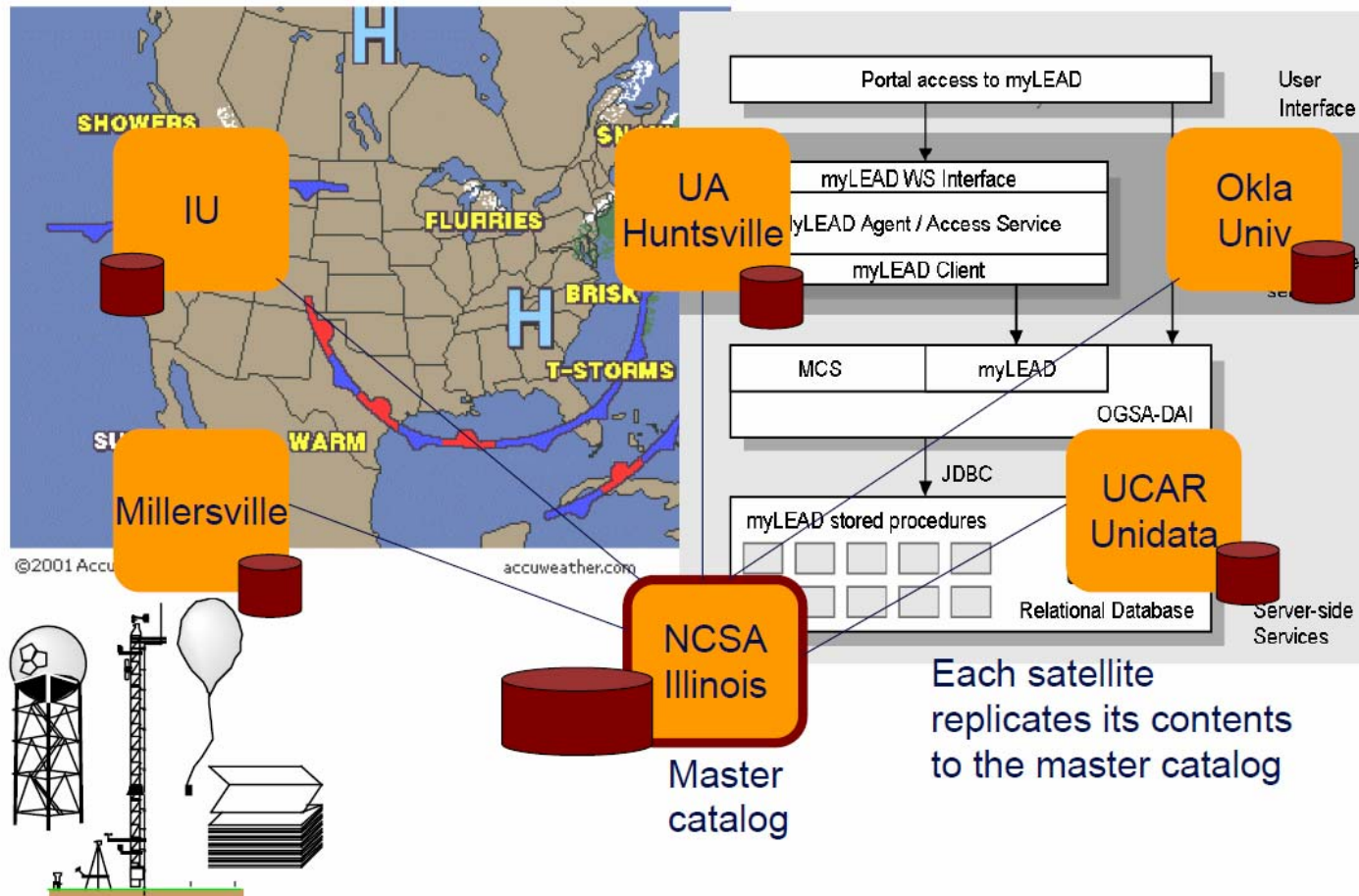
- Data&Computing Gridのショーケース

Sun Grid Engine + OGSA-DAI

- 利用者はデータをGTIに投入
- EPCCで解析、結果を表示(TOG)



LEAD: OGSA-DAIによる気象衛星データの管理



★ 応用の特徴

- 計算と分散データベース処理の連携
 - 例: 分散遺伝子DBを統合してクラスタ上でBlastにかけるとか。
 - Grid のミドルウェアに基づいている (GT4, OMII) ことが有利
- セキュリティ基盤との連携
 - GSIに基づく、グリッドの認証
 - VOMS ?
- 第3者データ転送、大量データ転送
 - 大量の検索結果がクライアントに直接戻るとまずい
 - 検索結果を別のサイトに送ったり、
 - GridFTPなどと結びつけたり、

OGSA-DAIの構造

OGSA-DAIの基本構造(1)

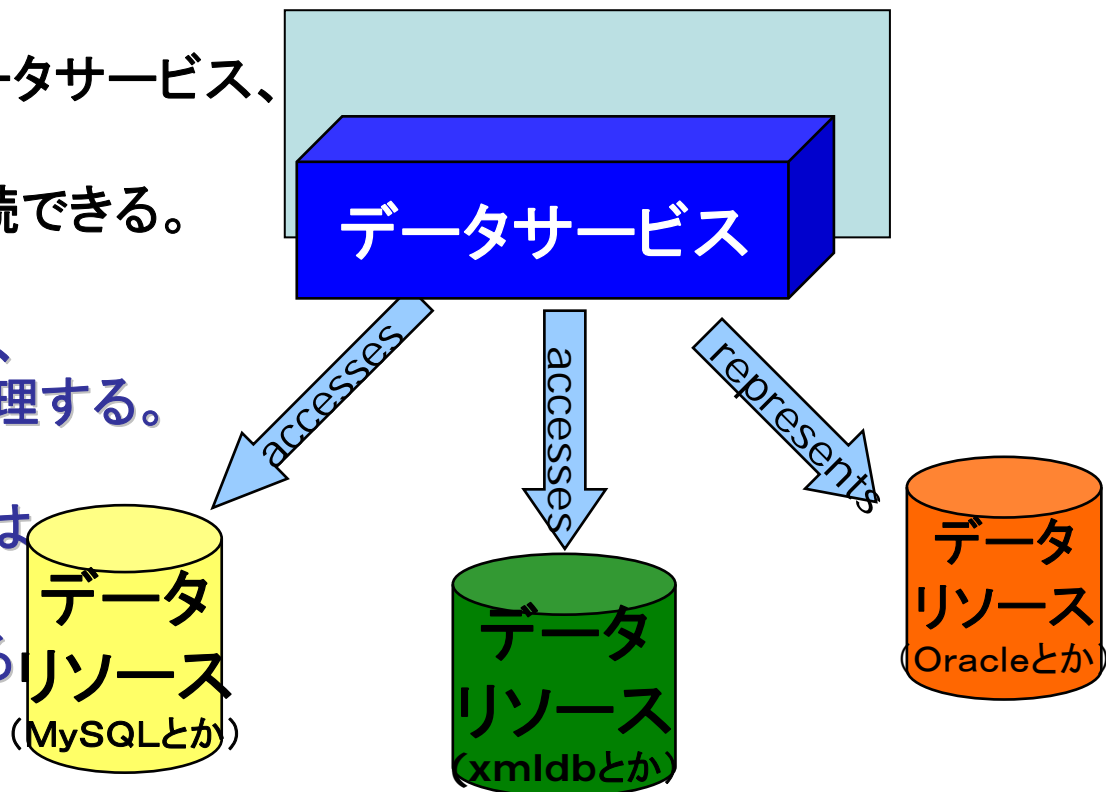
<http://localhost:8080/wsrf/ogsadai/DataService/DataResource>
/データサービス/データリソース

データサービス: OGSA-DAIの機能を提供するサービスの1インスタンス
データリソース: データサービスが管理するデータベース

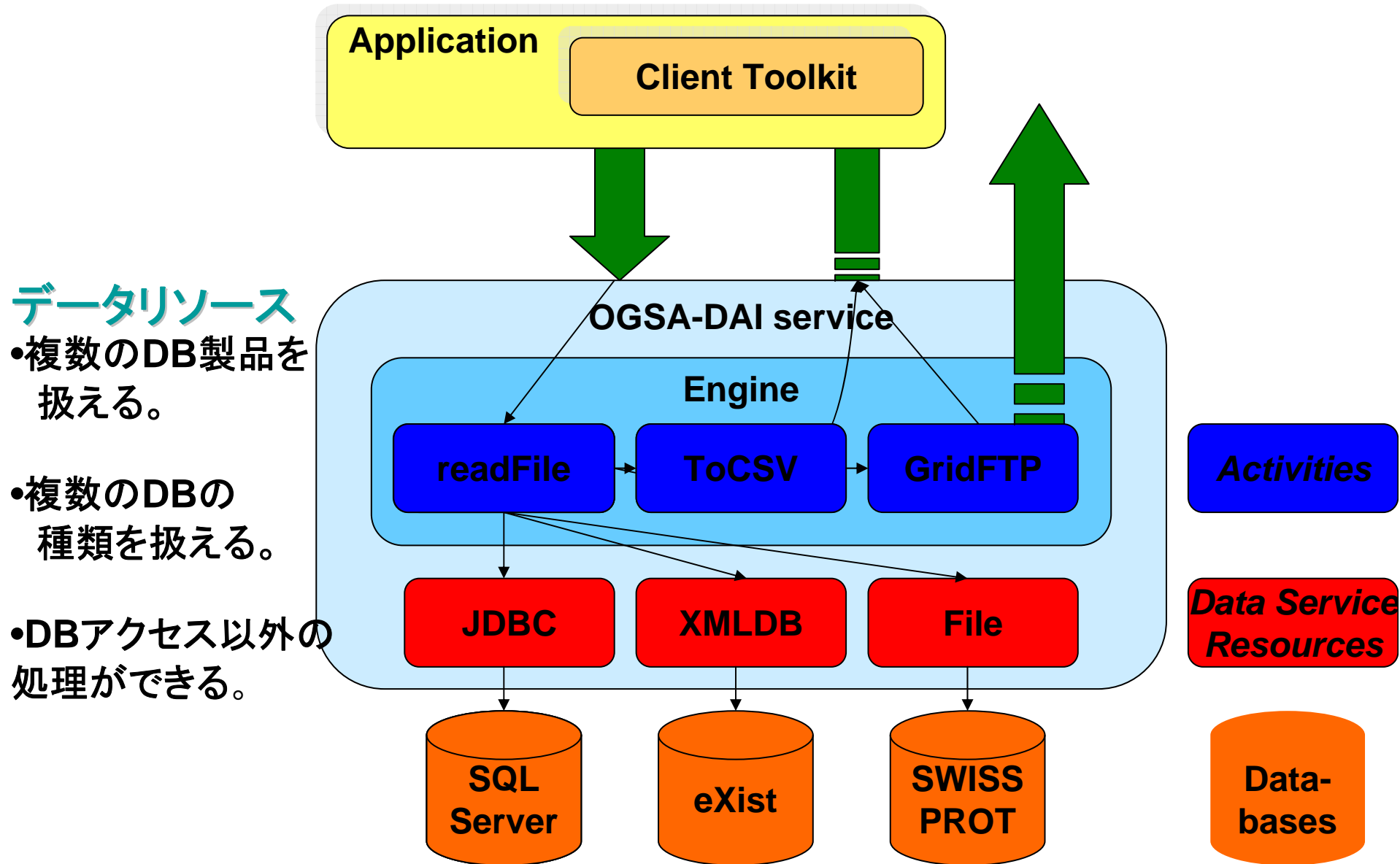
- 1つのogsadaiに複数のデータサービス、
- 1つのデータサービスに
複数のデータリソースが接続できる。

データサービスに要求を投げ、
データリソースをアクセス、処理する。

つまり、OGSA-DAIの導入とは
データサービスの導入と、
データリソースの導入からなる。



OGSA-DAIの基本構造(2)



データリソース

- 複数のDB製品を扱える。

- 複数のDBの種類を扱える。

- DBアクセス以外の処理ができる。

★ OGSA-DAIの構造(再掲): 分散処理へのアプローチ

● やりたいこと

- 分散データベース処理を含む分散ワークフロー
 - サービス連携
 - SOAベースの応用開発

ナイーブに応用を作ると性能が悪くなりそうだ。

★ OGSA-DAIにおける考え方。

1. 一つのデータサービスでの処理を高度化する。

- Activityの連携でワークフローを記述、処理
 - ひとつのサービス内でできることを高度化

2. 複数のサービスにおける連携(後述)

- データベース処理を含む汎用ワークフロー
 - BPELやTavernaなど、汎用ワークフローエンジンと組み合わせよ。
 - 該当モジュールの提供
 - WEEPなどDAI向きBPELエンジンの提供
- 分散データベース処理に基づくワークフロー
 - 分散問い合わせ処理のミドルウェア+ α の提供
 - OGSA-DQP
 - 問い合わせ処理を最適化する。

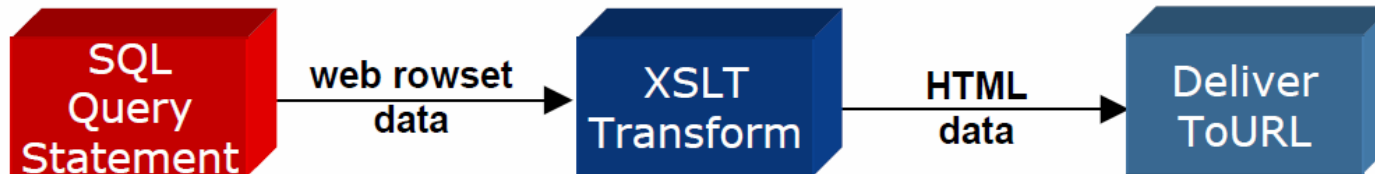
OGSA-DAIにおける処理

- PerformDocument
- Activity

この2つの要素が重要。

処理のモデル＝アクティビティ

1つの処理要求＝1連のアクティビティからなる。



例：DBを検索してXSLTで変換、結果をHTML化してURLの先に送る

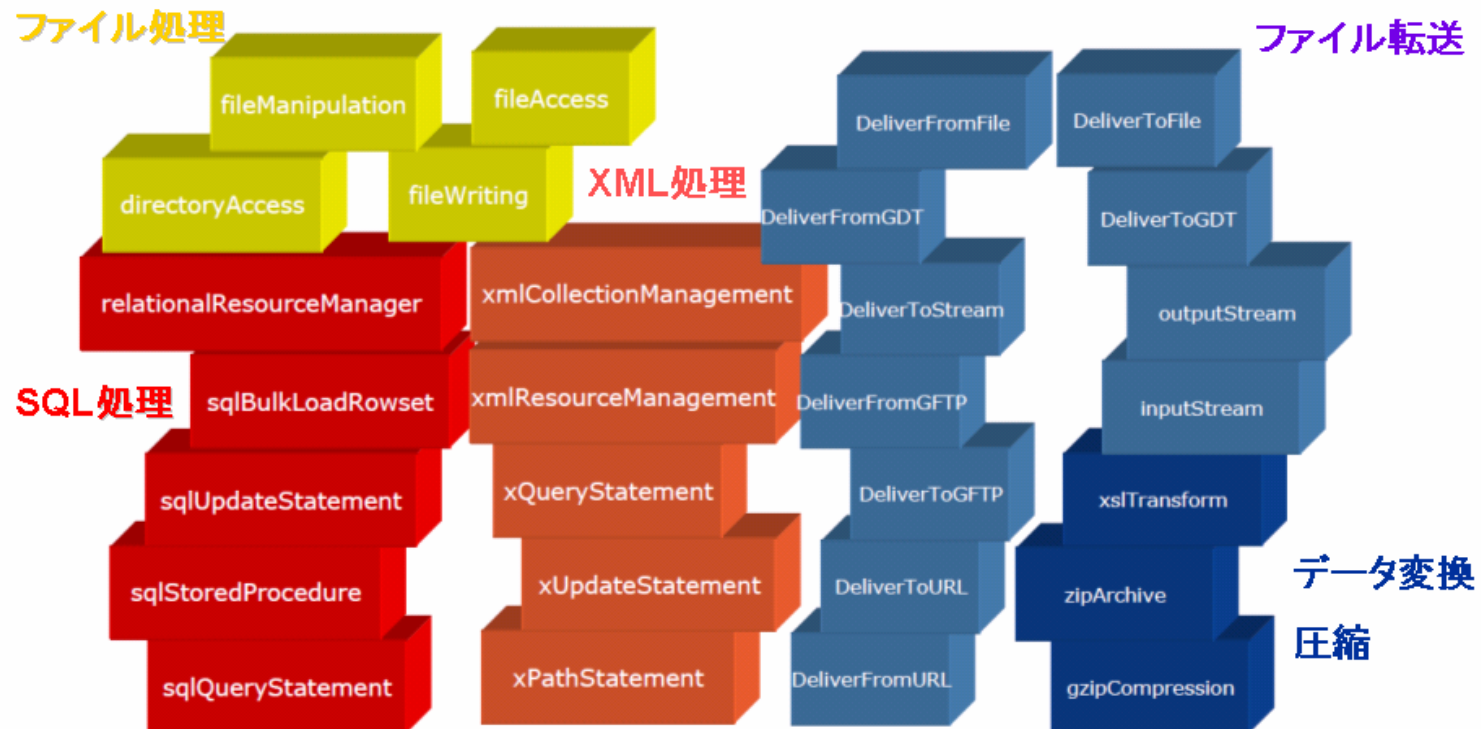
- 問い合わせ
 - SQL文の発行
 - XMLDBの場合は、XPathなど
 - 変換
 - 結果のXSLT変換： 検索結果 (Java rowset) の加工など
 - 転送
 - 結果の直接転送
 - FTPによる別のサイトへの転送
-
- アクティビティ間でデータ転送がされる。

Activityとは

アクティビティ:OGSA-DAIにおける基本実行要素

- たとえば、SQLの問い合わせ1つとか。
- 検索結果のファイルへの書き込み、とか

OGSA-DAI=たくさんのアクティビティの集まり

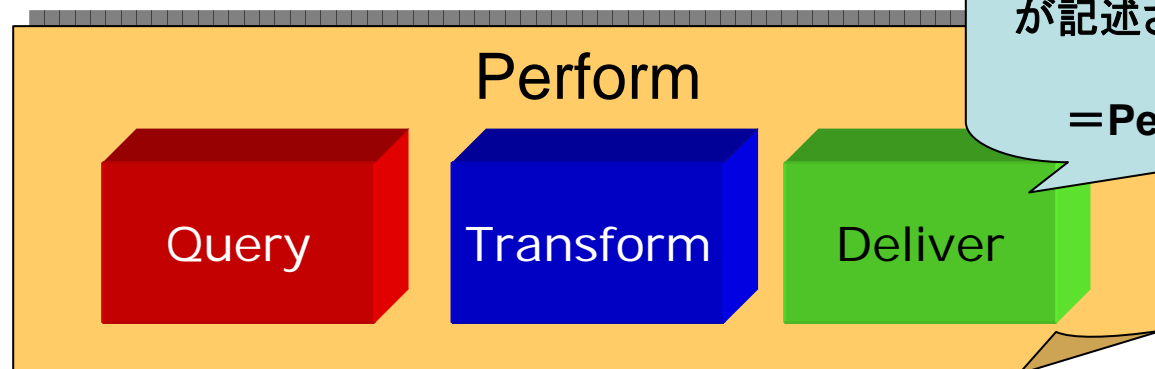


Perform Documentとは

- 複数のActivityがつながって、一連のデータ処理を記述するXMLドキュメント。



1. SQLで検索をして、(SQL問い合わせのアクティビティ)
2. 結果をXMLに変換して、(データ変換のアクティビティ)
3. そのデータをリモートに転送しよう、



これを、複数のアクティビティ
が記述された1枚のXMLファ
イルで書く
= PerformDocument

OGSA-DAIにおける処理

- 基本的には、各データサービスにPerformDocumentを作って投げる。
 - **PerformDocument**=簡単なデータ処理ワークフローが書ける。
(検索して、加工して、転送して、受け取って、変換して、、、などの1連の操作を1枚のXMLでかけるから)

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<perform xmlns="http://ogsadai.org.uk/namespaces/2005/10/types">
```

```
<documentation>This example performs a simple select statement to retrieve one row from the test database. The results are delivered within the response document.</documentation>
```

```
<sqlQueryStatement name="statement">
```

```
<expression>select * from littleblackbook where id=10</expression>
```

```
<resultStream name="statementOutputRS" />
```

```
</sqlQueryStatement>
```

```
<sqlResultsToXML name="statementRSToXML">
```

```
<resultSet from="statementOutputRS" />
```

```
<webRowSet name="statementOutput" />
```

```
</sqlResultsToXML>
```

```
</perform>
```

SQLで検索して
結果は
statementOutputRS

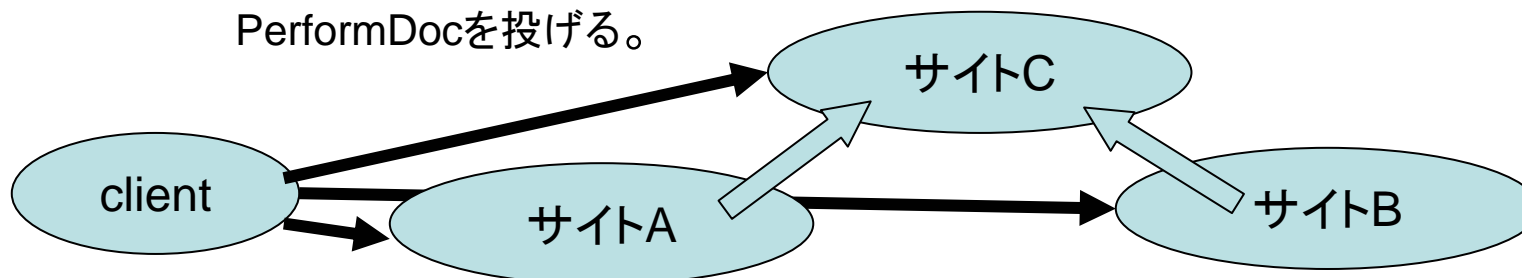
statementOutputRSストリームを、
RowSetに変換して出力

OGSA-DAIにおける処理(続)

- データ転送のアクティビティがあるから、分散処理ができる。
 - ややこしいネットワーク越しのデータ転送が簡単化、GridFTPもOK.
 - 第3者転送ができるから、連携もできる。
 - Aの検索結果をBに送れ、という要求を、Cから投げられる。
 - Bでは、Aからの転送結果を、たとえばファイルに書け、といったような記述ができる

サイト間の連携処理ができる。例えば、

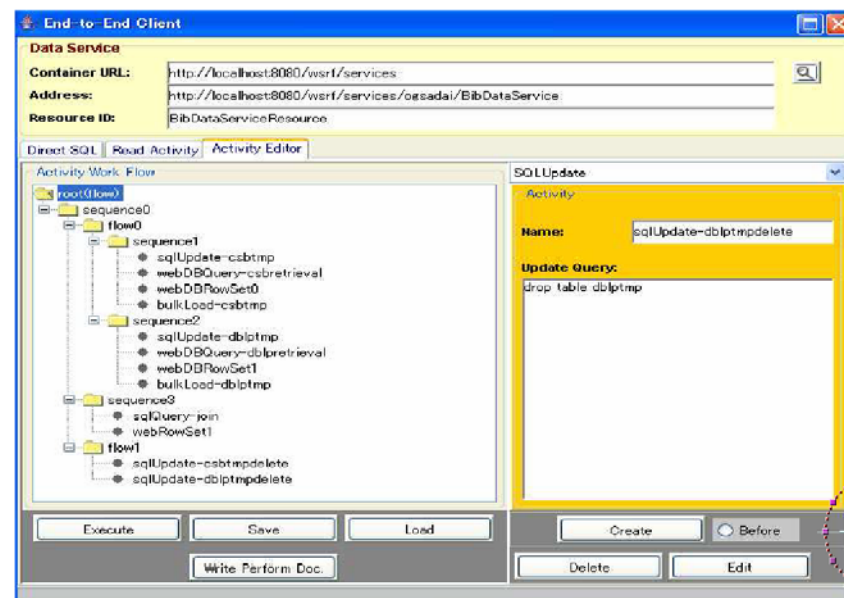
1. サイトAに、検索をして結果をCに送るPerformDocを投げる。
2. サイトBに、検索をして結果をCに送るPerformDocを投げる。
3. サイトCに、AとBからデータを受け取って一時テーブルに格納し、検索をして結果を返すPerformDocを投げる。



★ Activityのモデル

- 簡単な並列・順次実行コンポーネントを持つ
- Flow & Sequence
 - Flow要素のアクティビティは、並行実行が可能。
 - Sequence要素のアクティビティは、順次実行

AIST Activity Workflow Editor



javaプログラムからの実行

- いちいちPerformDocを書くのはどうも、、
 - Javaを使ってデータベースプログラムを書きたい。
- Client Toolkit** がある。

```
package clienttoolkit;
```

```
import uk.org.ogsadai.client.toolkit.GenericServiceFetcher;  
import uk.org.ogsadai.client.toolkit.Response;  
import uk.org.ogsadai.client.toolkit.activity.ActivityRequest;  
import uk.org.ogsadai.client.toolkit.activity.sql.SQLQuery;  
import uk.org.ogsadai.client.toolkit.activity.sql.WebRowSet;  
import uk.org.ogsadai.client.toolkit.service.DataService;
```

```
public class SimpleExample { public static void main(String[] args) throws Exception {  
    String handle = "http://localhost:8080/wsrf/services/ogsadai/DataService";  
    String id = "MySQLResource";  
    DataService service = GenericServiceFetcher.getInstance().getDataService(handle, id);  
    SQLQuery query = new SQLQuery("SELECT * FROM littleblackbook WHERE id='3475'");  
    WebRowSet rowset = new WebRowSet( query.getOutput() );  
    ActivityRequest request = new ActivityRequest();  
    request.add( query ); request.add( rowset );  
    Response response = service.perform( request );  
    System.out.println(response.getAsString()); }  
}
```

サポート・プラットフォーム

- Webサービス基盤
 - Globus Toolkit 4
 - Axis1.2.1
 - OMII
- データベース
 - 関係DB
 - MySQL, PostgreSQL, Oracle, DB2, SQLServer
 - XMLDB
 - eXist, Xindice,
 - File
 - Textファイル、CSV, binary, SWISSPROT, OMIM

拡張性

- **OGSA-DAI=Activityの集合**
Activity=XMLSchemaで書かれたIOインターフェイスと、実装(クラス)の記述からなる。

OGSA-DAIの構成ファイルは、Activityの定義の集まり。

Activityは拡張・変更できる

- 新しいインターフェイスと実装を追加する。
- 既存のインターフェイスを改造する。
- 実装を差し替える、
- などなど、、、

Activity定義・拡張の例

- GEODE project: (産業医学・健康管理)
 - DB周辺処理のアクティビティ化
 - ルール処理
 - メタデータ登録・管理など
 - マクロ的・典型的な処理のアクティビティ化
- 産総研 OGSA-WebDB,OGSA-DAI-RDF
<http://www.dbgrid.org/>
- OGSA-WebDB:
 - Webデータベースを仮想的に関係DBとしてみせるActivity
- OGSA-DAI-RDF:
 - Jena,SesameといったRDFのデータベース処理のActivity

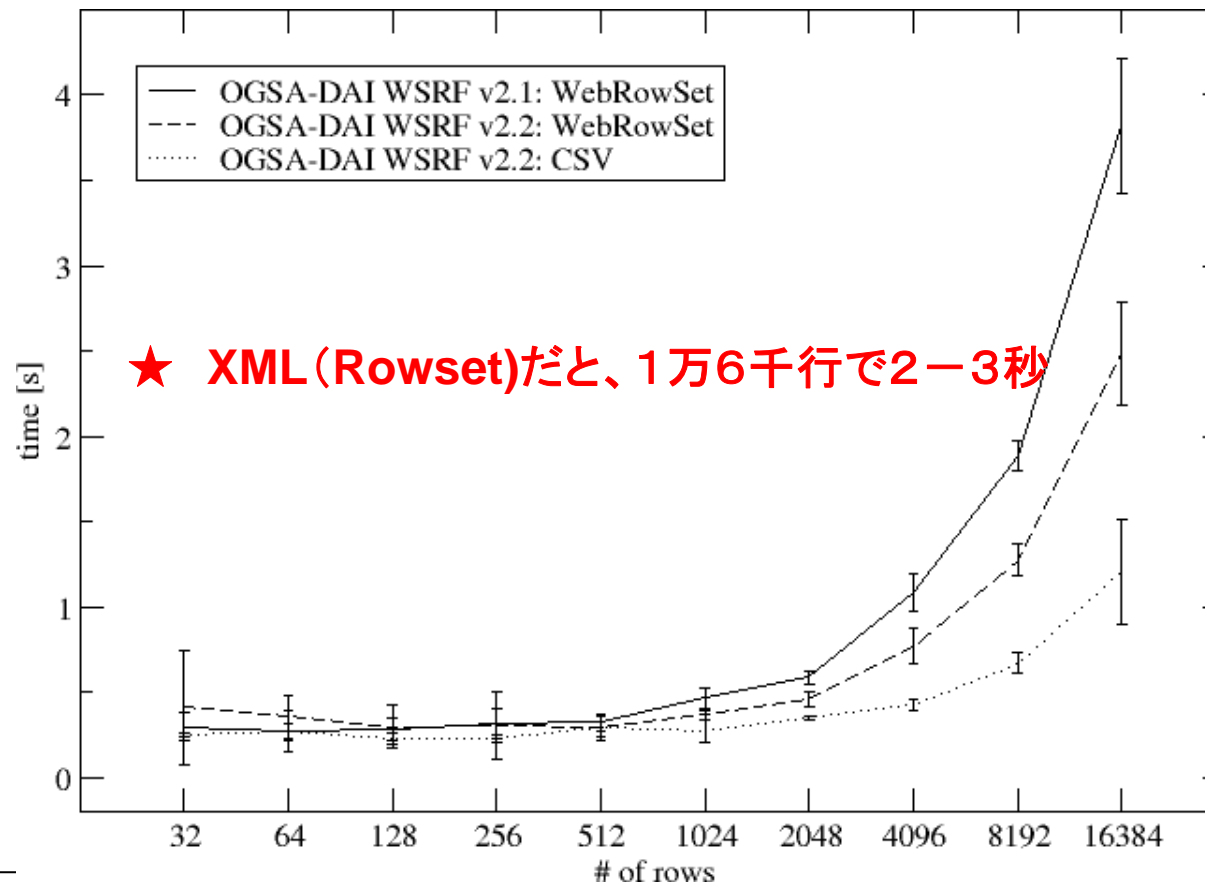


直近のバージョン 2.2

- 特徴
 - メモリ上の一時リソースの支援
 - リソースの動的な削除 (Withdraw)
 - マルチリソースの支援 (同じ問い合わせをみんなに投げる)
 - Activityの充実: BLOB, XQuery、メタデータ情報を取得するActivityなど
 - メッセージレベルセキュリティ
 - 性能の向上など

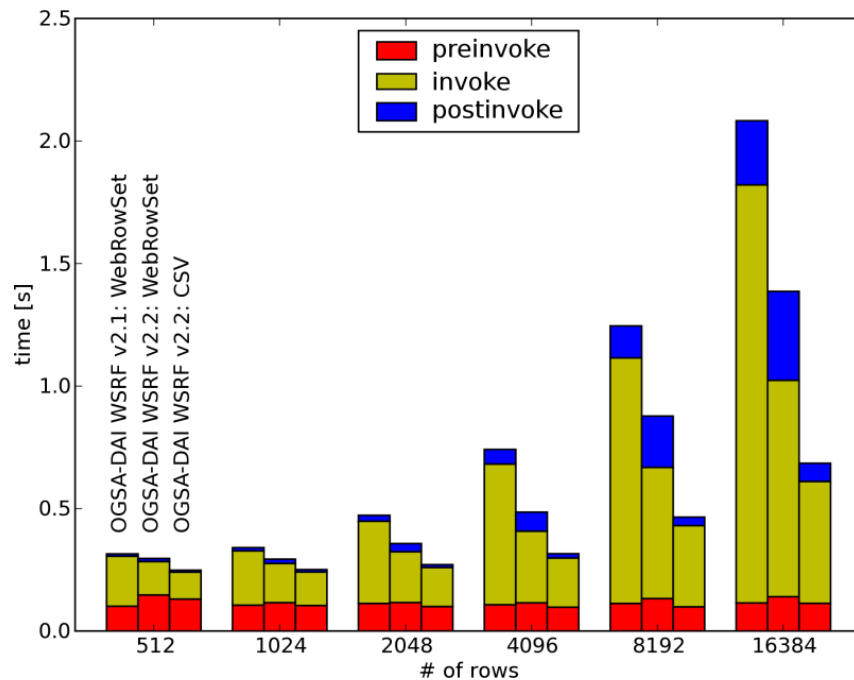
OGSA-DAIの性能

- サービスベース & XMLベースの転送
 - データ量が大きくなるに従って増加



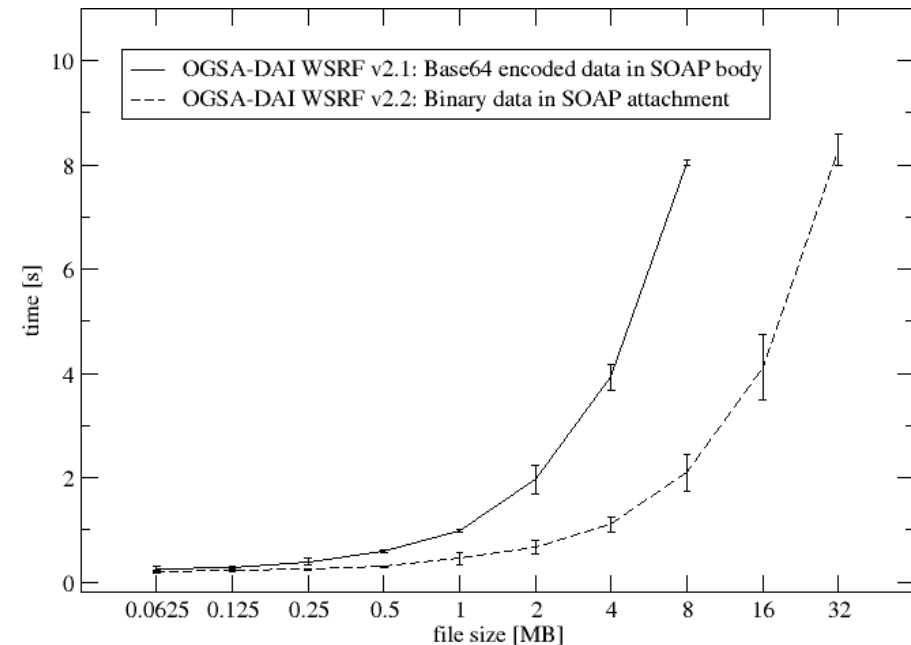
継続的な改良

- サーバソフトの改良



★ 緑色がサーバの処理時間

- バイナリ表現の利用による転送データサイズの縮小



★ 8MBの結果データで8秒→2秒

DAIS、WS-DAIとは (Database Access and Integration Service)

DAIS: Database Access and Integration Service

- DAIS-WG: OGFにおける、データベースアクセス標準の策定のためのワーキンググループ
 - WS-DAI: DB実装に非依存なデータベースアクセス規格
 - WS-DAIR: 関係データベースの接続規格
 - WS-DAIX: XMLDBの接続規格

規格のドキュメントとしては終了。

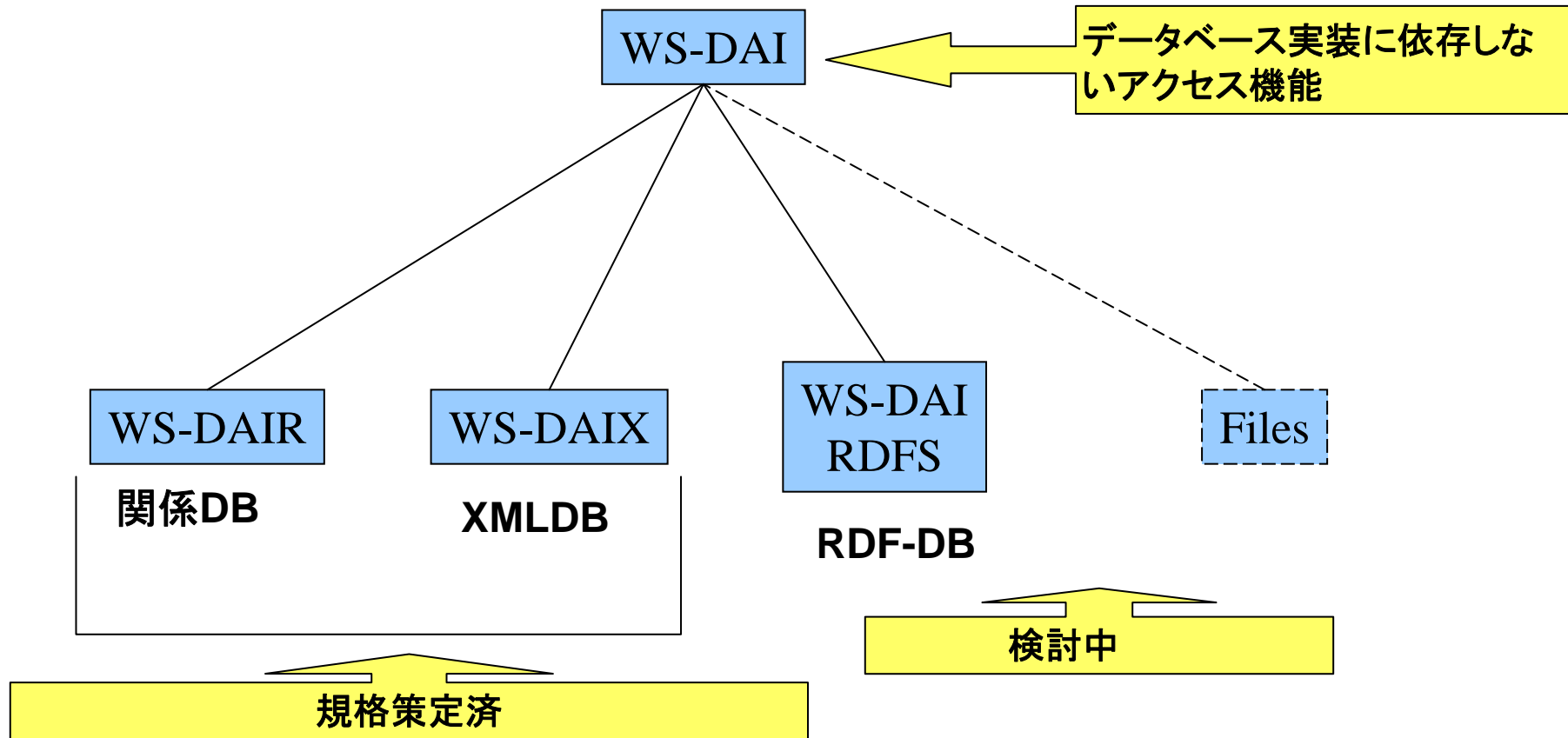
現在は互換性テストのための実装待ち

- ★ XML,RDBで各2つ、合計4つがいる、、
一つづつはOGSA-DAI系で実装できるが、、

WS-DAIの規格の動機と特徴

- Webサービスに基づいたデータベースアクセス規格
 - SQLを渡すにしても標準化されたインターフェイスが必要、、、
 - 意外とない。
- いろいろなデータベースをなるべく統一的に扱いたい
 - RDB, XMLDB, RDF, ファイル、などなど、、
- 大量データを扱えるようにしたい。
 - 検索結果が直接クライアントに戻っては困る場合がある。

DAIS 規格の構造

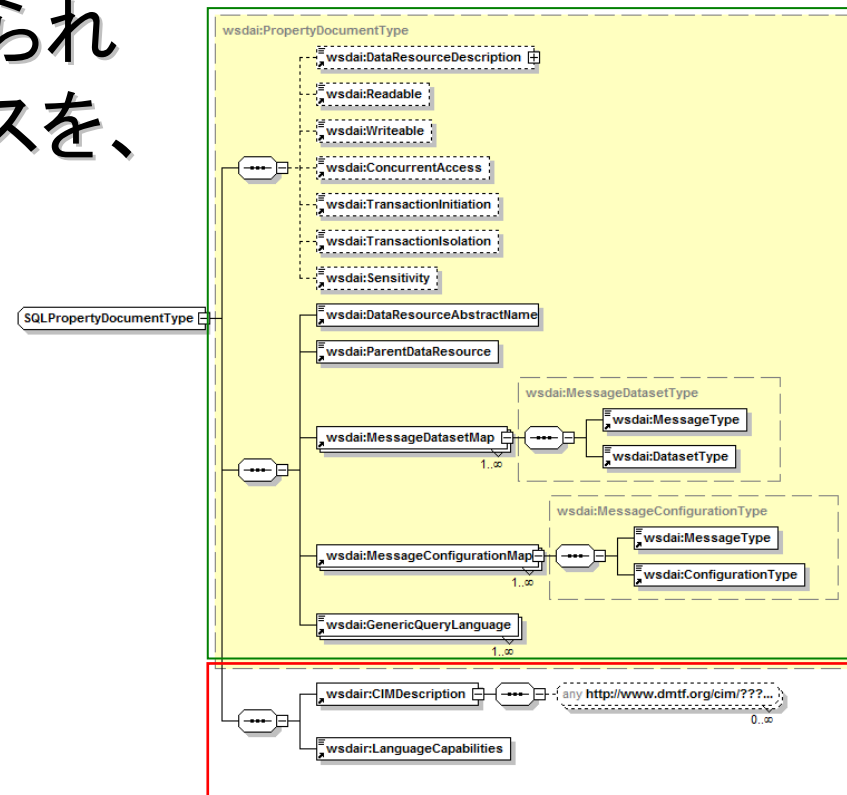


WS-DAIと WS-DAIR,X など下位規格との関係

- 基本的に、WS-DAIで定められたメッセージインターフェイスを、

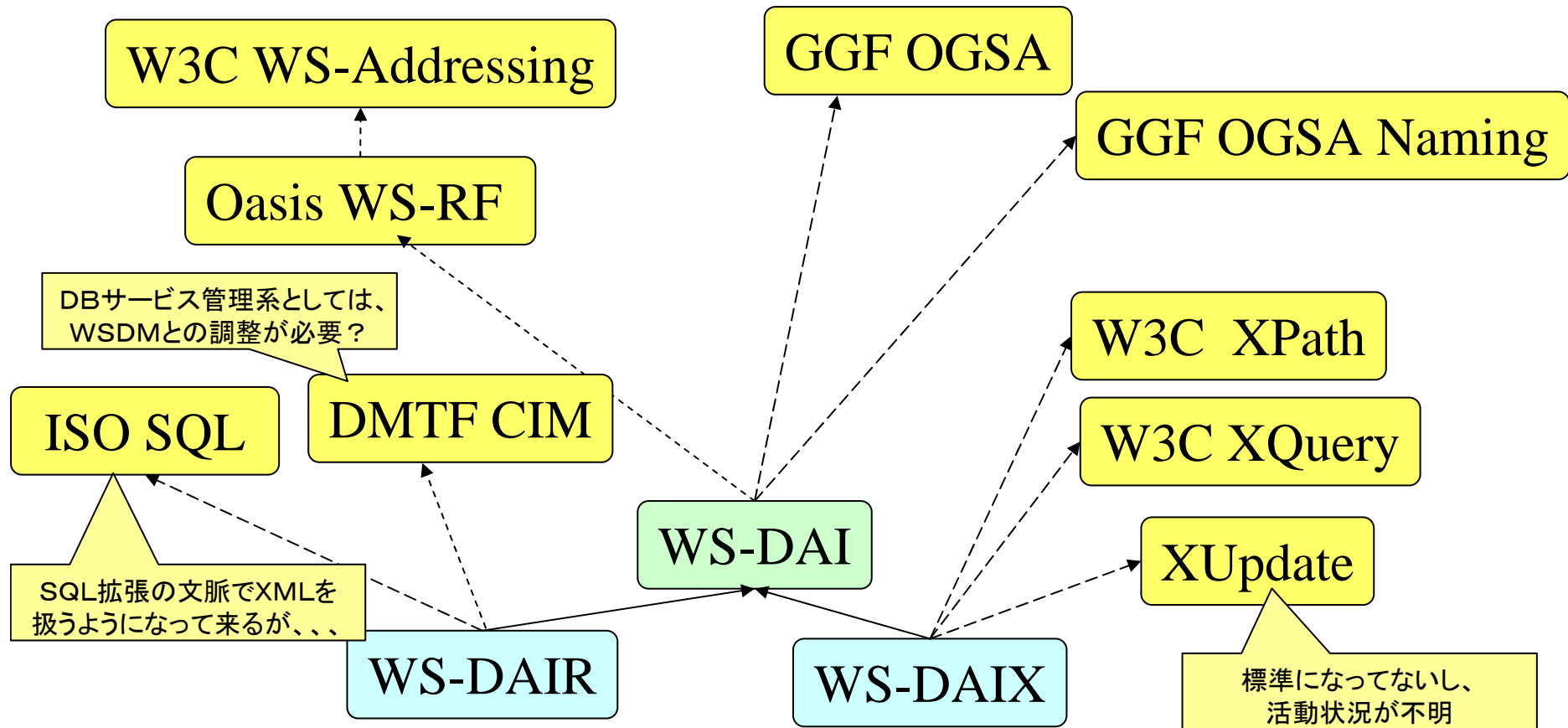
個々のDB実装に対応して
拡張あるいは詳細化する。

WS-DAIで決まる規格



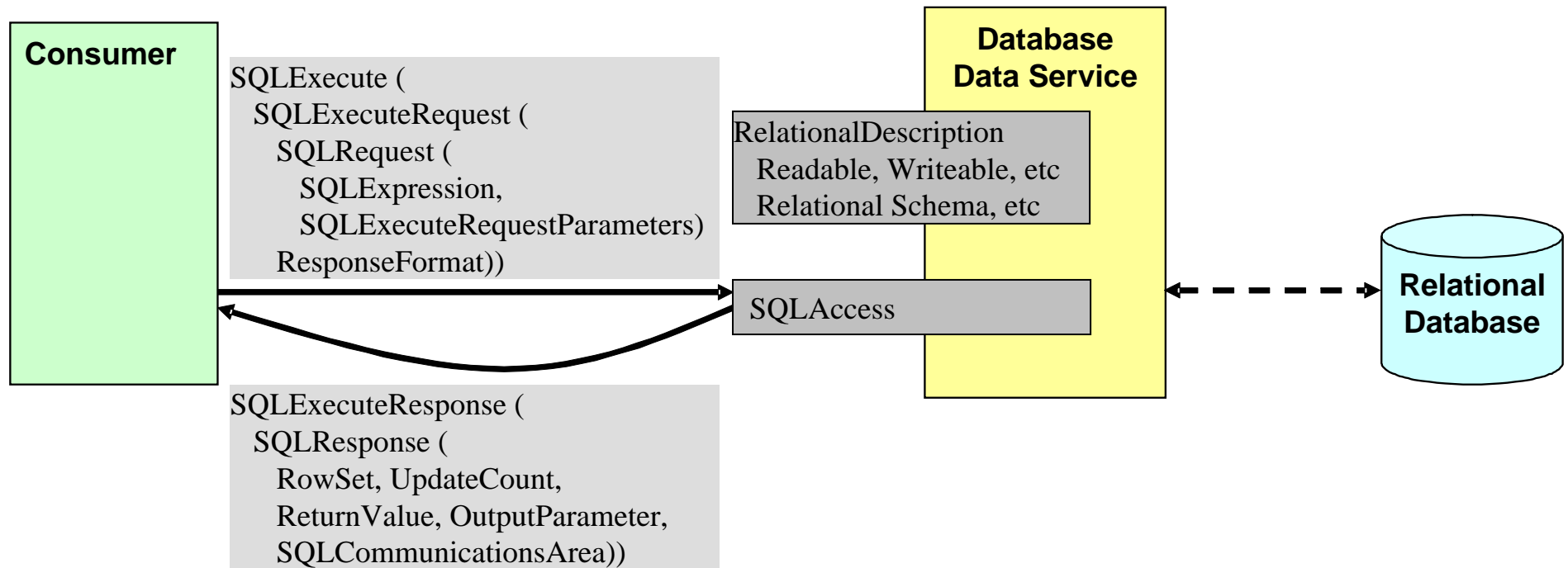
WS-DAIRなど、下位の規格の拡張

関連する規格

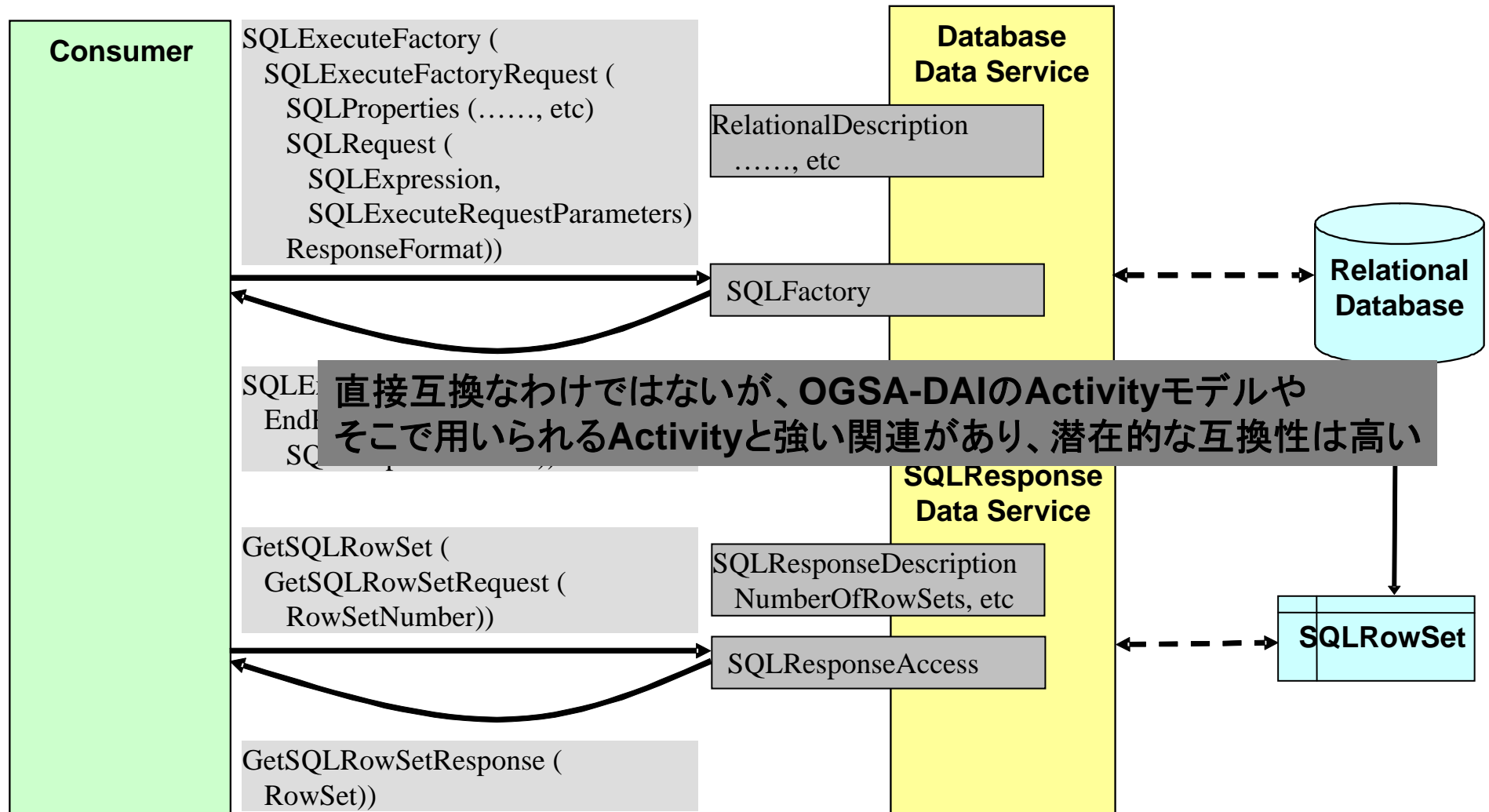


DAISアクセスパターン1: 直接配送

普通のクライアントサーバにおけるSQL処理



DAIS 2:間接配送:データ量が多いときなど、検索結果を集合として、別のサービスから提供する



★ DAISのモデルの特徴(まとめ)

- 結果の保持と別サービスによる提供
 - 大量の結果データへの対応
 - 第三者転送への対応
- 複数のデータベース(RDB, XML等)に対応する共通モデル。
 - どこに何があるかとかは、知りやすい。
- 標準の採用
 - CIMの利用、DMTFなどとの連携も含む

DAIS仕様とOGSA-DAIの関連

- DAIS Specification
 - WS-DAI
 - WS-DAIR(Relational)
 - WS-DAIX(XMLDB)
- OGSA
- 準拠する規格
 - WS-DAIはWS-RF非依存！
 - WS-RFへのマッピング
 - WS-Iのマップの可能性

- OGSA-DAI
 - 現在は非対応(仕様のには親和性)
 - 共通インターフェイス
 - Relational
 - XML
- 現時点ではOGSA準拠ではない
 - OGSAの仕様が後に策定
- 複数の版がある。
 - Globus Toolkit(WS-RF)4.X版
 - WS-I, OMII 版。

WS-DAI規格の拡張性

- WS-DAIR(Relational), DAIX(XML)は、コアなWS-DAI規格の拡張である。
- 他のデータベースについても容易に拡張できる。
 - Files: ボランティア募集中だが、人がいない。
 - Object指向DB: ラフな提案が出たが活動中断状態
 - 標準言語の欠如、利用人口やマーケットの小ささなど。
 - RDF DB: 産総研とEu Ontogridが共同で提案/作業中
 - W3C SPARQLなどをベースにグリッド向け拡張
 - Ontology Primitivesの提供
 - ユースケース & ボランティア募集中

OGSA-DQPとは

分散処理へのアプローチ

- やりたいこと

- 分散データベース処理を含む分散ワークフロー
 - サービス連携
 - SOAベースの応用開発

ナイーブに応用を作ると性能が悪くなりそうだ。

OGSA-DAIにおける考え方。

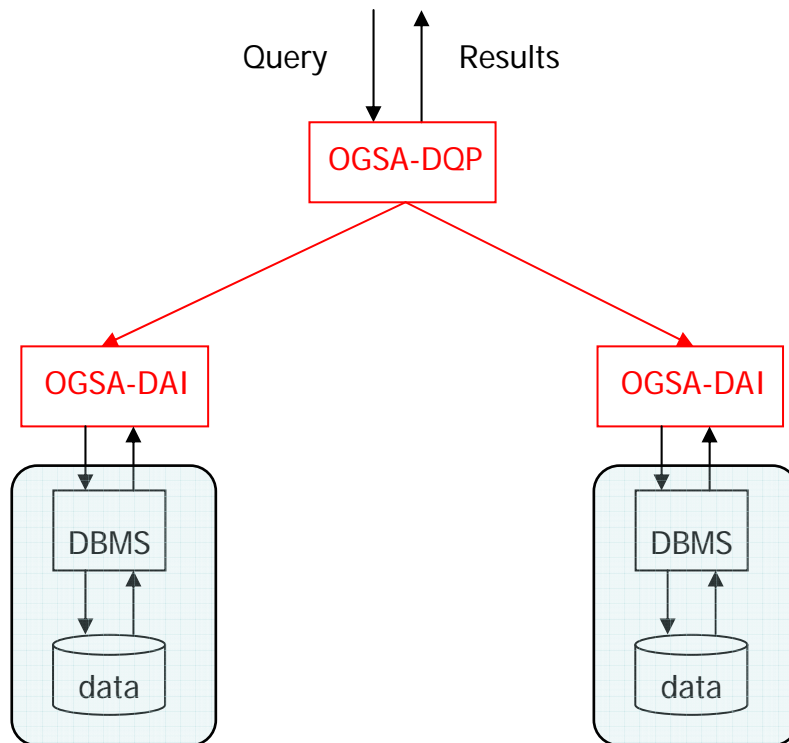
1. 一つのデータサービス内でのワークフロー

- Activityの連携でワークフローを記述、処理
 - ひとつのサービス内でできることを高度化

2. 複数のサービスにおける連携

- データベース処理を含む汎用ワークフロー
- BPELやTavernaなど、汎用ワークフローエンジンと組み合わせよ。
 - 該当モジュールの提供
 - WEEPなどDAI向きBPELエンジンの提供
- 分散データベース処理に基づくワークフロー
 - 分散問い合わせ処理のミドルウェア+ α の提供
 - OGSA-DQP
 - 問い合わせ処理を最適化する。

OGSA-DQP (Distributed Query Processing) とは



- OGSA-DAIの関連プロジェクト.
- 分散SQLのミドルウェア
- 複数のOGSA-DAIを連携、統合。
 - ただしSQLのみ
 - Webサービスを呼べる
- DQP自身もOGSA-DAI互換サービスとして実装。
- 問い合わせ最適化
 - スケジューリング
 - 並行実行
 - データ転送

SQLの例

- Given two DBMSs and one analysis tool (i.e., a Web service):
 - **goTerm** : MySQLで作られたGene Ontology のデータベース。OGSA-DAI が実装されている。ice
 - **protein** : 別のDBの protein sequence DB, これも OGSA-DAI で実装。service
 - **Blast** (Web サービス経由で実行できるものとする);
- 分散した複数のDBを結合して、その結果をBlastにかける。

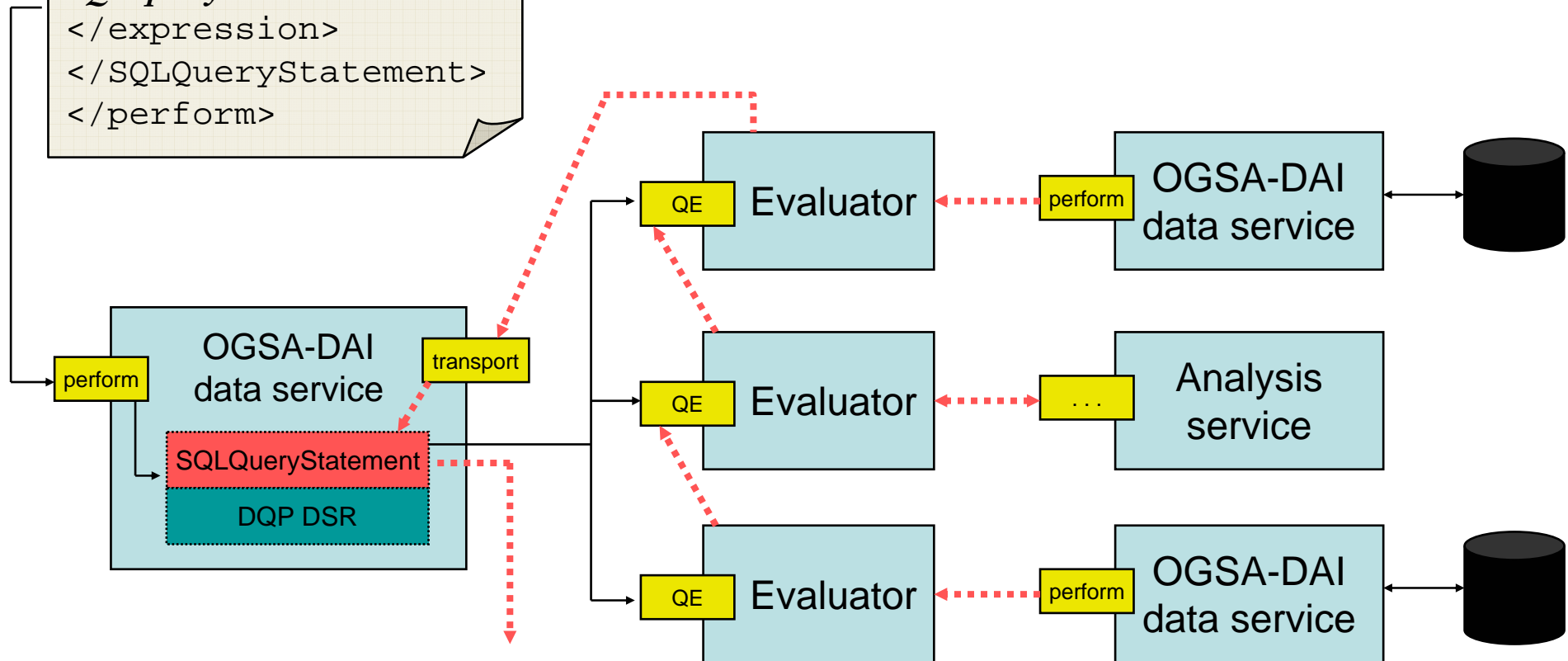
```
select p.proteinId, Blast(p.sequence)
from protein p, goTerm t
where t.termId = 'GO:0005942' and p.proteinId=t.proteinId
```

DQPの処理概要

Activityの連携(各サイトにおける処理とデータ転送の指定)を、うまく使って処理を分割・指示できる。

```

<perform>
<SQLQueryStatement>
<expression>
SQL query
</expression>
</SQLQueryStatement>
</perform>
    
```



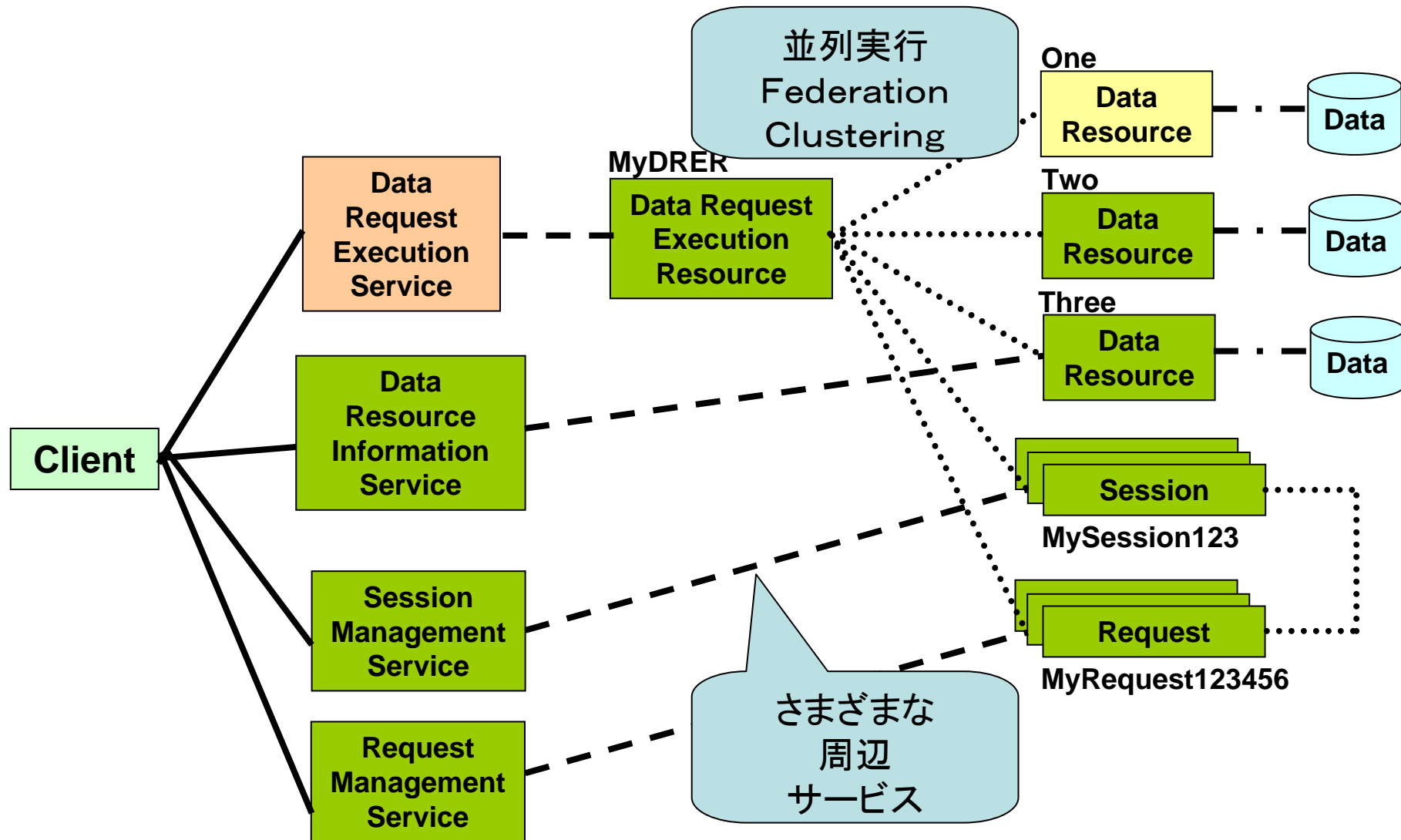
Result: WebRowSet XML Stream

OGSA-DAI 3.0

OGSA-DAI WSRF/OMII 3.0

- 作り直し: 2007年の6月に出る予定。
- 主な特徴
 - Multi-Resource Requests (複数のデータリソースに同じ処理を投げられる)
→ *data integration capabilities* (統合がやさしくなる)
 - Expanded Resource Model (セッションやリクエスト、実行状態の管理)
→ *standardised, separate concerns*
 - Persistence (システム状態の管理)
→ *robustness, dynamic reconfiguration*
 - Clustering (データベースのクラスタリングをサポート)
→ *scalability, robustness*
 - New Activity Framework (Activityのモデルが変わり、Federationが可能)
→ *simpler, more powerful*
 - Monitoring Framework (モニタリングやロギングの機能の拡大)
→ *logging, auditing, status reporting*

OGSA-DAI 3.0の構造概要



★ 詳細なスケジュール & 協力体制等

スケジュール

- 6月末: Beta
 - WSRF (GT4) and AXIS (OMII)
 - By OMII-UK
- Unicore/gLite版
 - By OMII-Europe
 - Work With VOMS

AIST (産総研) & OGSA-DAI & OGF DAIS

密接な関係にある

DAIS-WG

OGSA-DAI-RDF/WebDB

DQP

Geogrid

OMII/Uk-e-Science <http://www.omii.ac.uk/>

OMII Successful Story (WebDB with BST)

OMI PAL (開発中コードへのアクセス等の権限)

アジア・パシフィックではAISTのみ

まとめ

OGSA-DAI <http://www.ogsadai.org.uk/>

- OGF規格WS-DAIと連携したミドルウェア
 - WS-DAI,DAIR,DAIX
 - Webサービスに基づくDBアクセスの規格
- Activity&PerformDocument
 - 一つのDBサービス内でのワークフロー
 - 第3者転送など。
 - OGSA-DAI3.0がまもなく出る。
- DQP
 - 複数のデータサービス間での分散処理